A Preliminary Conceptualization and Analysis on Automated Static Analysis Tools for Vulnerability Detection in Android Apps

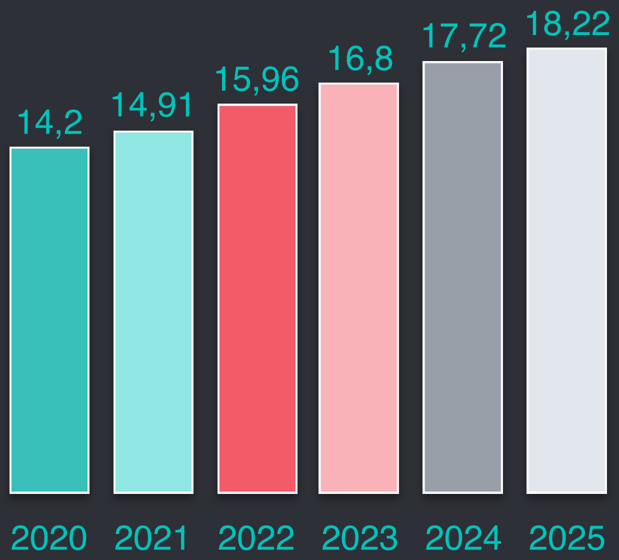
Giammaria Giordano, Fabio Palomba, Filomena Ferrucci

University of Salerno (Italy)
Software Engineering (SeSa) Lab
Department of Computer Science

✉ giagiordano@unisa.it
🌐 https://broke31.github.io/giammaria-giordano/
GiammariaGiord1

sesa lab
SOFTWARE ENGINEERING SALERNO

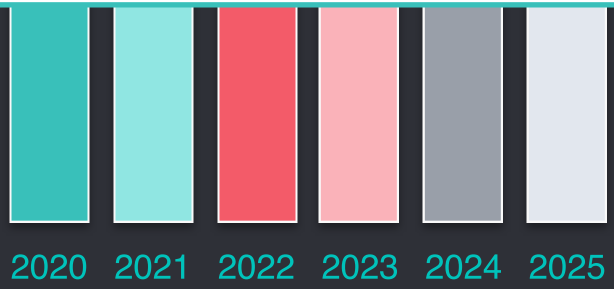
Number of Mobile Devices Worldwide from 2020 to 2025 (in billions)



https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/

Number of Mobile Devices Worldwide from 2020 to 2025 (in billions)

80% of mobile devices are Android devices



https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/

Number of Mobile Devices Worldwide from 2020 to 2025 (in billions)

80% of mobile devices are Android devices

Current World Population is **8 billions**

https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/

2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)

An Empirical Assessment of Security Risks of Global Android Banking Apps

Sen Chen[1], Lingling Fan[1], Guozhu Meng[2,3], Ting Su[4], Minhui Xue[5], Yinxing Xue[6]
Yang Liu[1,8], Lihua Xu[7]
[1]Nanyang Technological University, Singapore
[2]SKLOIS, Institute of Information Engineering, Chinese Academy of Sciences, China
[3]School of Cyber Security, University of Chinese Academy of Sciences, China [4]ETH Zurich, Switzerland
[5]The University of Adelaide, Australia [6]University of Science and Technology of China, China
[7]New York University Shanghai, China [8]Zhejiang Sci-Tech University, China
chensen@ntu.edu.sg

2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)

Automated Third-Party Library Detection for Android Applications: Are We There Yet?

Xian Zhan*
The Hong Kong Polytechnic University, Hong Kong, China
chichoxian@gmail.com

Lingling Fan*
College of Cyber Science, Nankai University, China
Nanyang Technological University, Singapore
ecnujanefan@gmail.com

Tianming Liu
Monash University
Australia
tianming.liu@monash.edu

Sen Chen
College of Intelligence and Computing, Tianjin University, China
Nanyang Technological University, Singapore
ecnuchensen@gmail.com

Li Li
Monash University
Australia
li.li@monash.edu

Haoyu Wang
Beijing University of Posts and Telecommunications
China
haoyuwang@bupt.edu.cn

Yifei Xu
Southern University of Science and Technology, China
1611209@mail.sustech.edu.cn

Xiapu Luo
The Hong Kong Polytechnic University, Hong Kong, China
luoxiapu@gmail.com

Yang Liu
Nanyang Technological University, Singapore
yangliu@ntu.edu.sg

Vulnerability Analysis of Android Auto Infotainment Apps

Amit Kr Mandal
Università Ca' Foscari Venezia, Italy
amitmandal.nitdgp@gmail.com

Agostino Cortesi
Università Ca' Foscari Venezia, Italy
cortesi@unive.it

Pietro Ferrara
JuliaSoft Srl, Verona, Italy
pietro.ferrara@juliasoft.com

Federica Panarotto
University of Verona, Italy
federica.panarotto@gmail.com

Fausto Spoto
University of Verona, Italy
fausto.spoto@univr.it

tudy of Static Analysis Tools to Detect Vulnerabilities of Branchless Banking Applications in Developing Countries

Fahad Ibrar*
Information Technology University
fahad.ibrar@itu.edu.pk

Hamza Saleem†
Information Technology University
hamza.saleem@itu.edu.pk

Sam Castle
The University of Washington
stcastle@cs.washington.edu

Muhammad Zubair Malik
Information Technology University
zubair.malik@itu.edu.pk

Although the vastness of proposed tools, we noticed a lack of empirical evaluation on the real capability of these static analysis tools to detect vulnerabilities

Research Questions

RQ1 - What are the **vulnerability types** identified by existing automated static analysis tools for mobile apps?

RQ2 - What are the capabilities of existing automated static analysis tools in terms of mobile **app analyzability**, **frequency of detection**, and **complementarity** among them?

How did we address the RQs?

For the first RQ, we manually extracted a **taxonomy of risks**

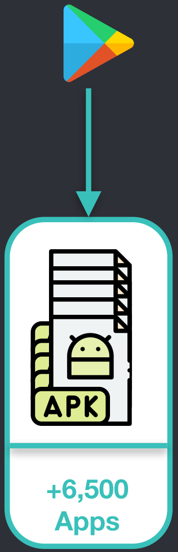
How did we address the RQs?

For the first RQ, we manually extracted a **taxonomy of risks**

For the second RQ, we analyzed the tools from a qualitative point of view by analyzing the **frequencies of risk detection** and the **complementarity** among them

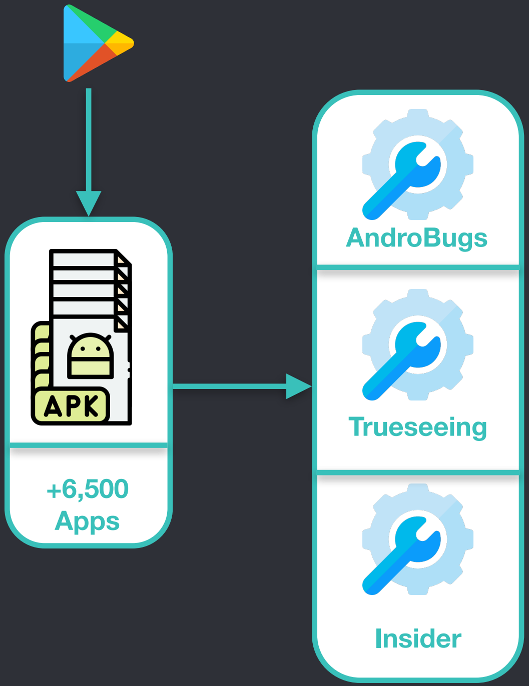
Research Method

Research Method

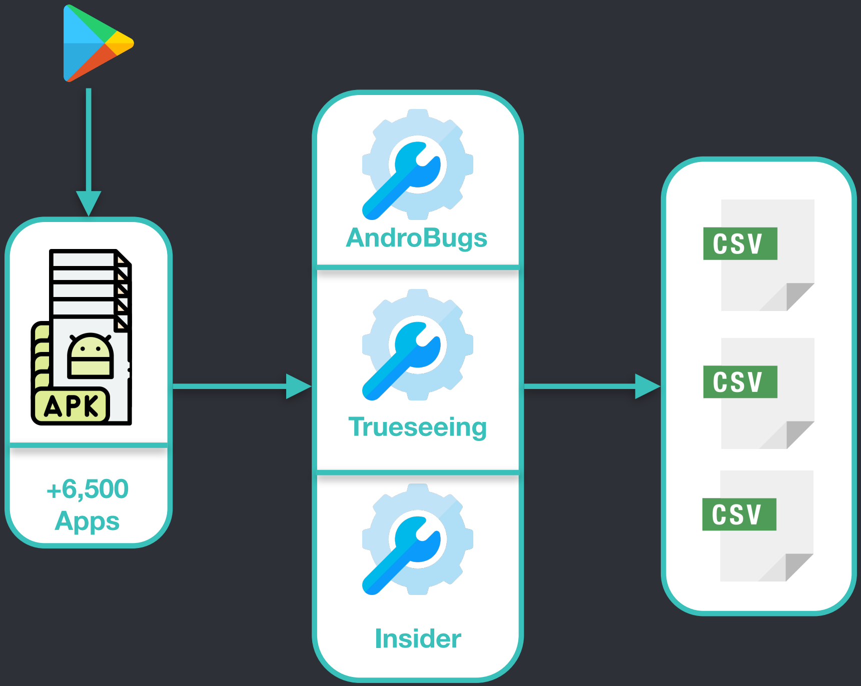


+6,500 Apps

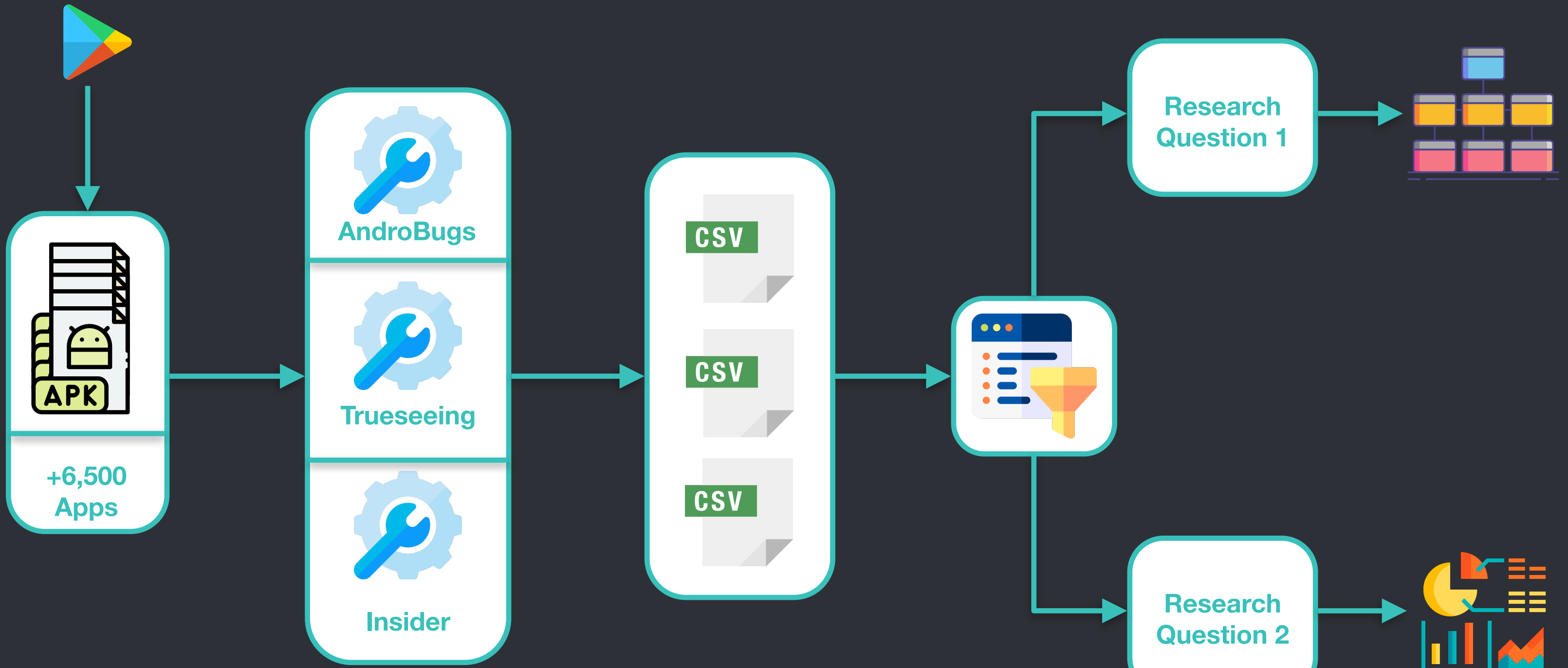
Research Method



Research Method



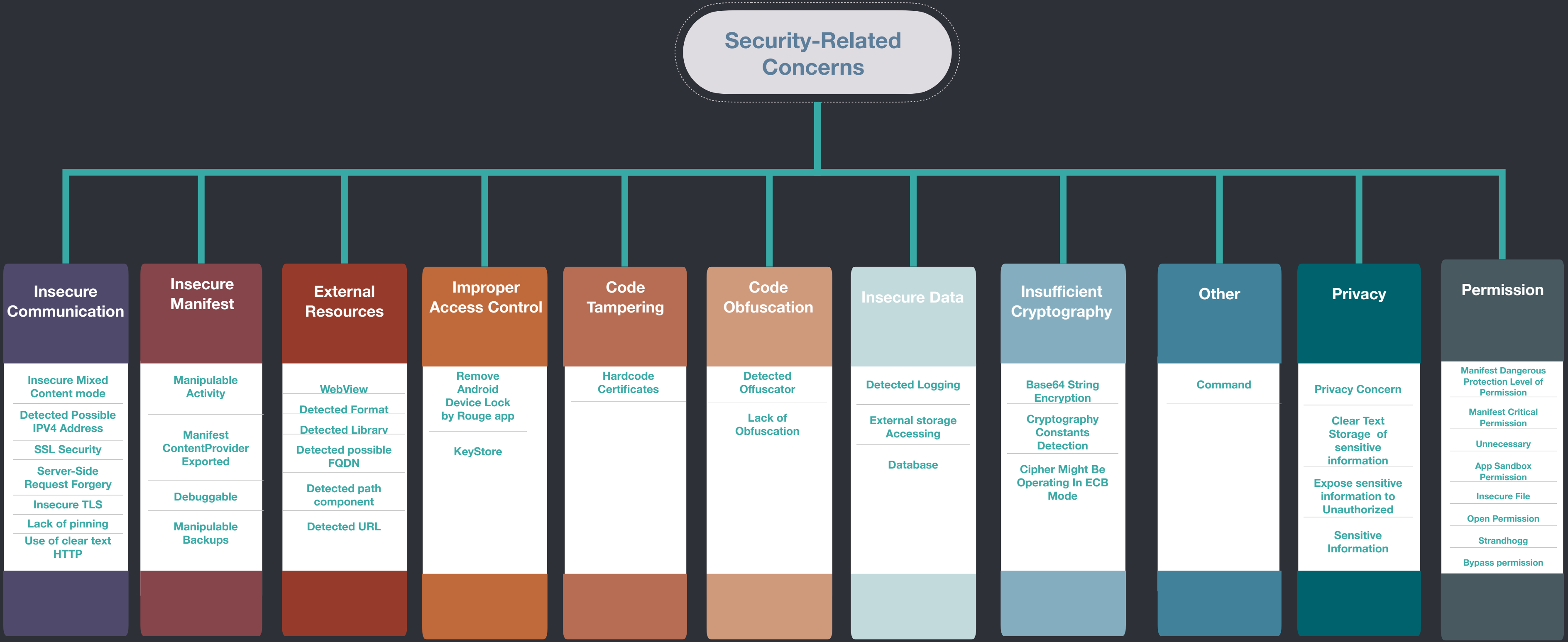
Research Method



Research Questions

RQ1 - What are the **vulnerability types** identified by existing automated static analysis tools for mobile apps?

RQ2 - What are the capabilities of existing automated static analysis tools in terms of mobile app **app analyzability**, **frequency of detection**, and **complementarity** among them?



Security-Related Concerns

Insecure Communication
- Insecure Mixed Content mode
- Detected Possible IPV4 Address
- SSL Security
- Server-Side Request Forgery
- Insecure TLS
- Lack of pinning
- Use of clear text HTTP

Insecure Manifest
- Manipulable Activity
- Manifest ContentProvider Exported
- Debuggable
- Manipulable Backups

External Resources
- WebView
- Detected Format
- Detected Library
- Detected possible FQDN
- Detected path component
- Detected URL

Improper Access Control
- Remove Android Device Lock by Rouge app
- KeyStore

Code Tampering
- Hardcode Certificates

Code Obfuscation
- Detected Offuscator
- Lack of Obfuscation

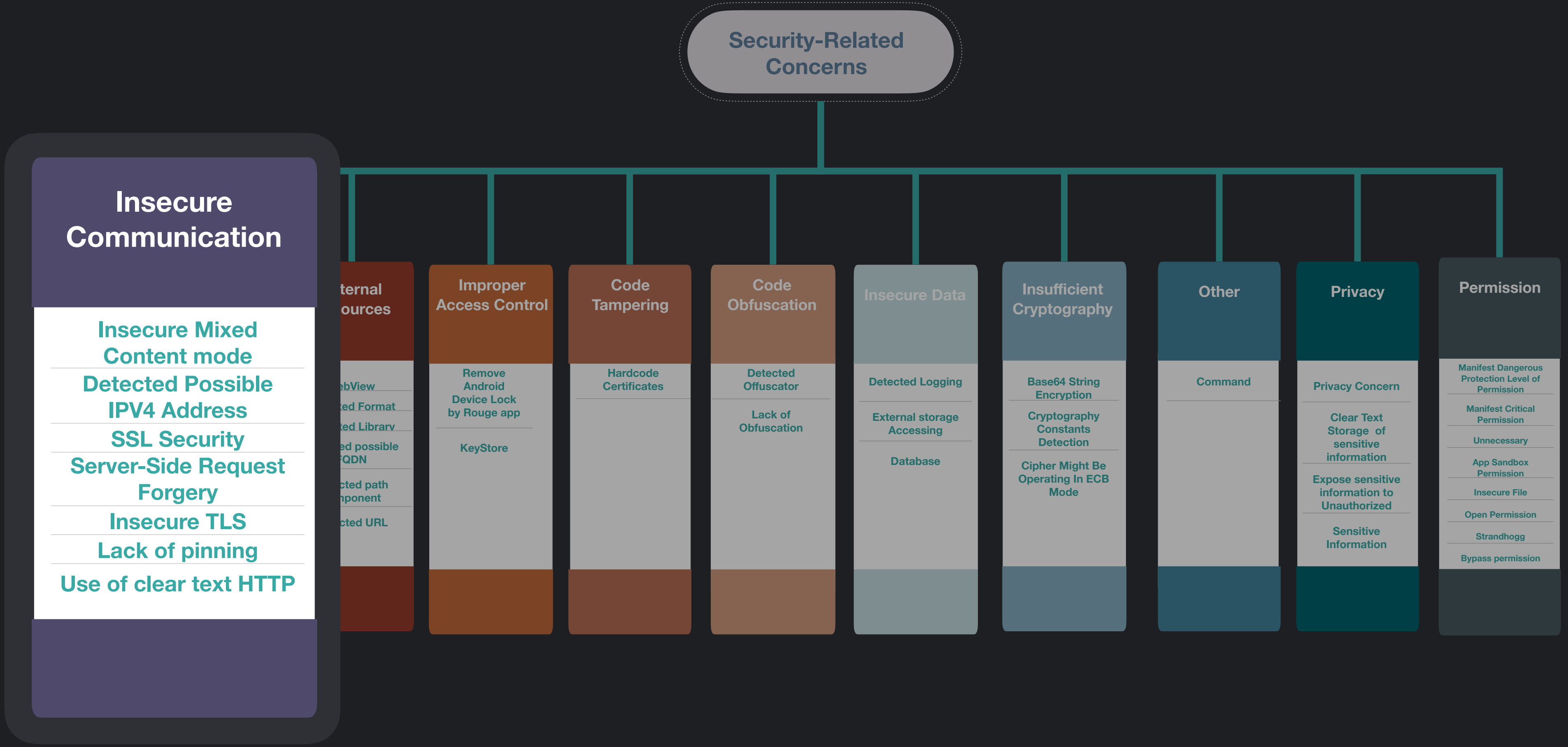
Insecure Data
- Detected Logging
- External storage Accessing
- Database

Insufficient Cryptography
- Base64 String Encryption
- Cryptography Constants Detection
- Cipher Might Be Operating In ECB Mode

Other
- Command

Privacy
- Privacy Concern
- Clear Text Storage of sensitive information
- Expose sensitive information to Unauthorized
- Sensitive Information

Permission
- Manifest Dangerous Protection Level of Permission
- Manifest Critical Permission
- Unnecessary
- App Sandbox Permission
- Insecure File
- Open Permission
- Strandhogg
- Bypass permission



Security-Related Concerns

Insecure Communication

- Insecure Mixed Content mode
- Detected Possible IPV4 Address
- SSL Security
- Server-Side Request Forgery
- Insecure TLS
- Lack of pinning
- Use of clear text HTTP

ternal ources

- ebView
- ted Format
- ted Library
- ed possible FQDN
- cted path mponent
- cted URL

Improper Access Control

- Remove Android Device Lock by Rouge app
- KeyStore

Code Tampering

- Hardcode Certificates

Code Obfuscation

- Detected Offuscator
- Lack of Obfuscation

Insecure Data

- Detected Logging
- External storage Accessing
- Database

Insufficient Cryptography

- Base64 String Encryption
- Cryptography Constants Detection
- Cipher Might Be Operating In ECB Mode

Other

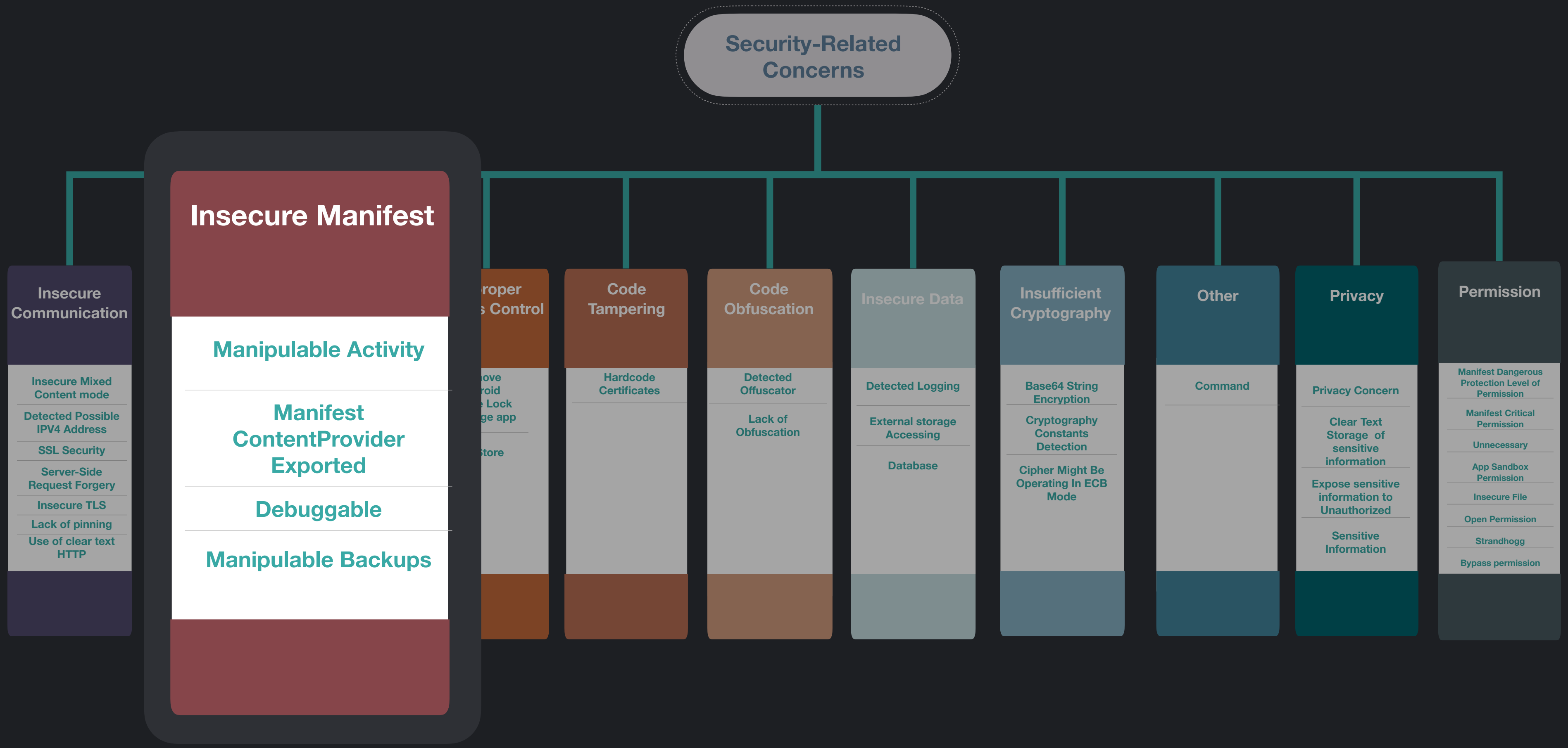
- Command

Privacy

- Privacy Concern
- Clear Text Storage of sensitive information
- Expose sensitive information to Unauthorized
- Sensitive Information

Permission

- Manifest Dangerous Protection Level of Permission
- Manifest Critical Permission
- Unnecessary
- App Sandbox Permission
- Insecure File
- Open Permission
- Strandhogg
- Bypass permission



Security-Related Concerns

Insecure Communication
- Insecure Mixed Content mode
- Detected Possible IPV4 Address
- SSL Security
- Server-Side Request Forgery
- Insecure TLS
- Lack of pinning
- Use of clear text HTTP

Insecure Manifest
- Manipulable Activity
- Manifest ContentProvider Exported
- Debuggable
- Manipulable Backups

roper s Control
- ove roid Lock ge app
- tore

Code Tampering
- Hardcode Certificates

Code Obfuscation
- Detected Offuscator
- Lack of Obfuscation

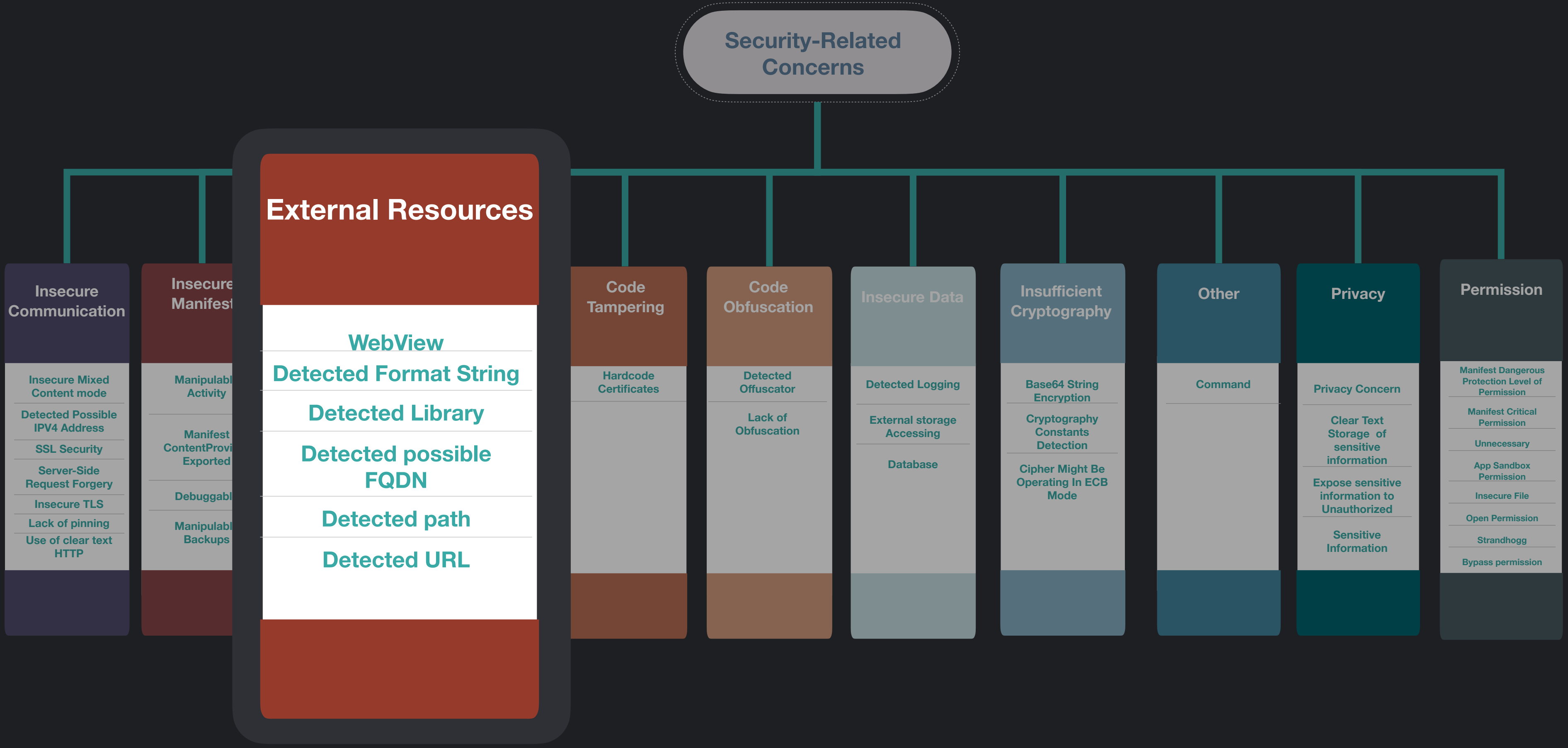
Insecure Data
- Detected Logging
- External storage Accessing
- Database

Insufficient Cryptography
- Base64 String Encryption
- Cryptography Constants Detection
- Cipher Might Be Operating In ECB Mode

Other
- Command

Privacy
- Privacy Concern
- Clear Text Storage of sensitive information
- Expose sensitive information to Unauthorized
- Sensitive Information

Permission
- Manifest Dangerous Protection Level of Permission
- Manifest Critical Permission
- Unnecessary
- App Sandbox Permission
- Insecure File
- Open Permission
- Strandhogg
- Bypass permission



Security-Related Concerns

Insecure Communication

Insecure Mixed Content mode

Detected Possible IPV4 Address

SSL Security

Server-Side Request Forgery

Insecure TLS

Lack of pinning

Use of clear text HTTP

Insecure Manifest

Manipulabl Activity

Manifest ContentProvi Exported

Debuggabl

Manipulabl Backups

External Resources

WebView

Detected Format String

Detected Library

Detected possible FQDN

Detected path

Detected URL

Code Tampering

Hardcode Certificates

Code Obfuscation

Detected Offuscator

Lack of Obfuscation

Insecure Data

Detected Logging

External storage Accessing

Database

Insufficient Cryptography

Base64 String Encryption

Cryptography Constants Detection

Cipher Might Be Operating In ECB Mode

Other

Command

Privacy

Privacy Concern

Clear Text Storage of sensitive information

Expose sensitive information to Unauthorized

Sensitive Information

Permission

Manifest Dangerous Protection Level of Permission

Manifest Critical Permission

Unnecessary

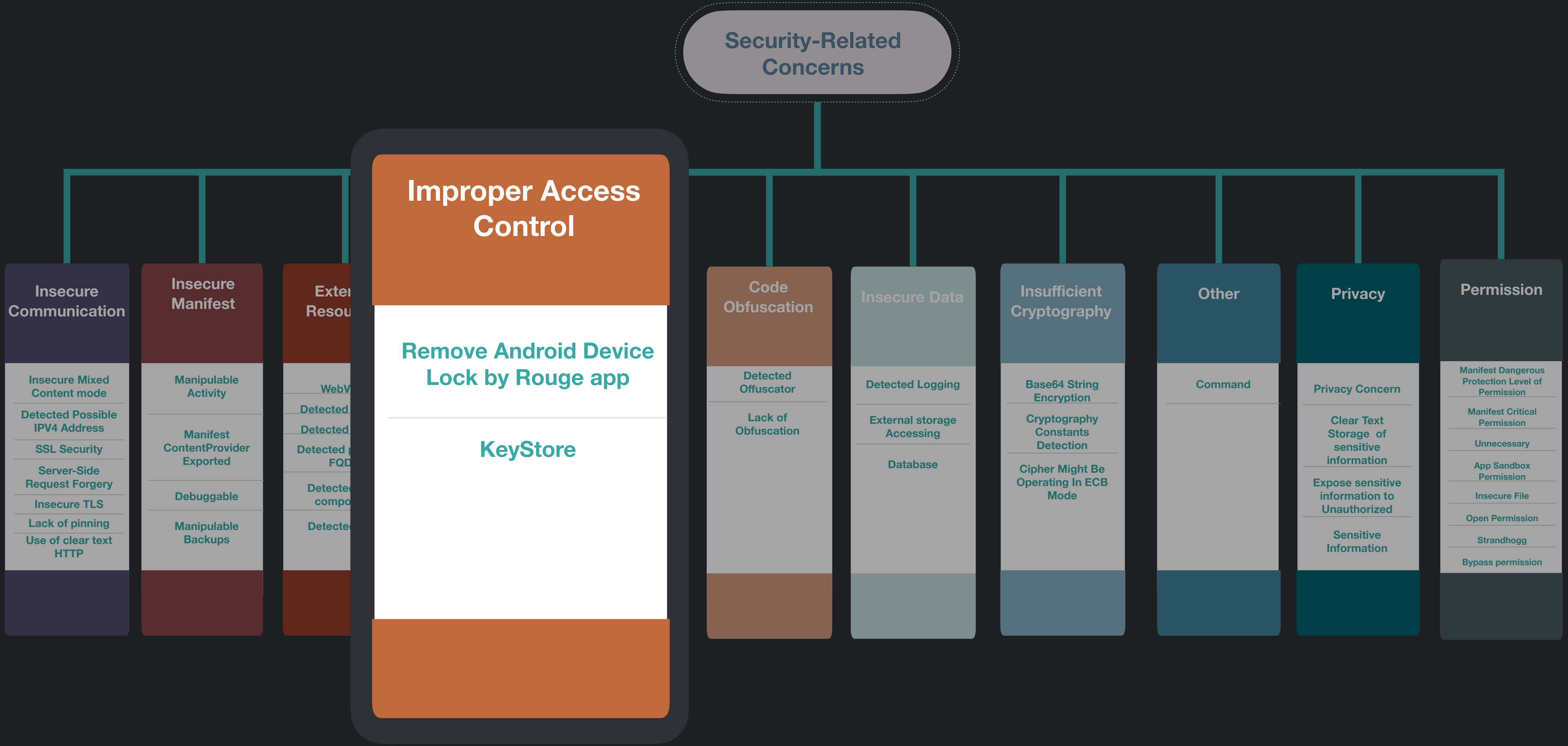
App Sandbox Permission

Insecure File

Open Permission

Strandhogg

Bypass permission



Security-Related Concerns

Insecure Communication
- Insecure Mixed Content mode
- Detected Possible IPV4 Address
- SSL Security
- Server-Side Request Forgery
- Insecure TLS
- Lack of pinning
- Use of clear text HTTP

Insecure Manifest
- Manipulable Activity
- Manifest ContentProvider Exported
- Debuggable
- Manipulable Backups

Exter Resou
- WebV
- Detected
- Detected
- Detected FQD
- Detecte compo
- Detecte

Improper Access Control
- Remove Android Device Lock by Rouge app
- KeyStore

Code Obfuscation
- Detected Offuscator
- Lack of Obfuscation

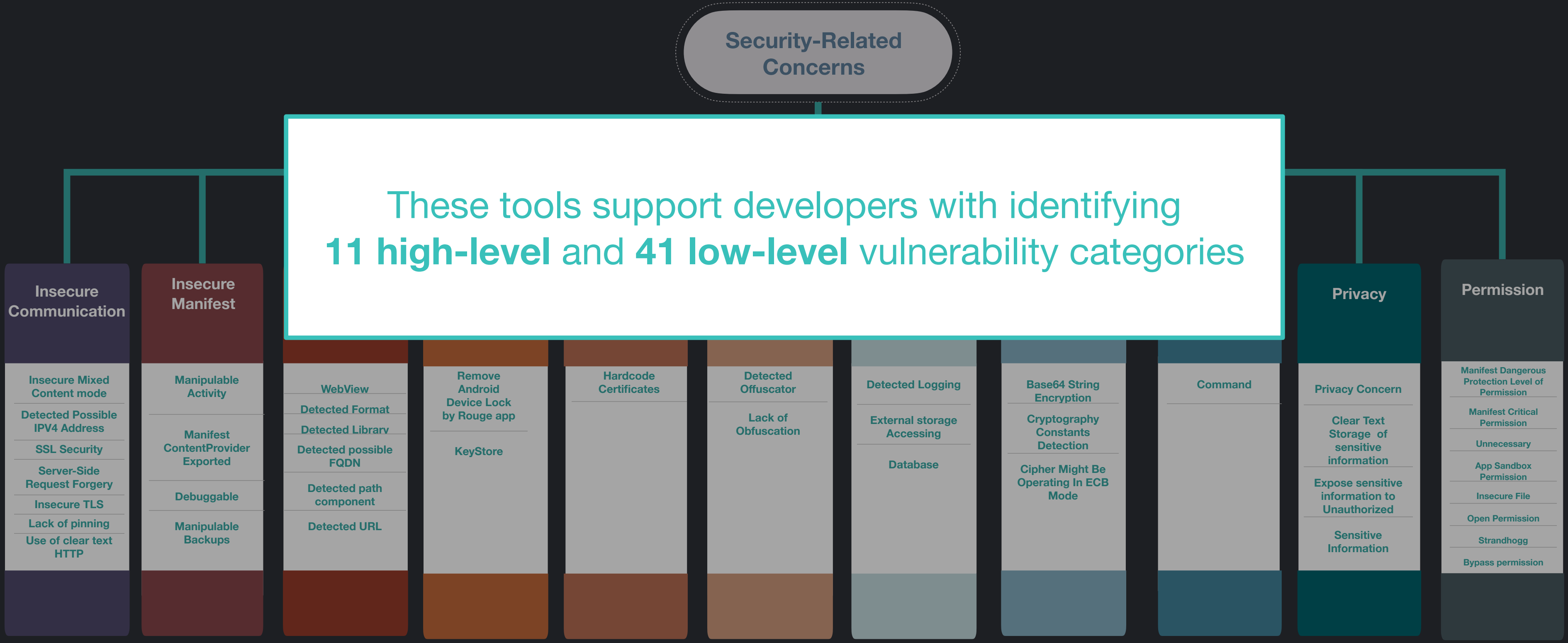
Insecure Data
- Detected Logging
- External storage Accessing
- Database

Insufficient Cryptography
- Base64 String Encryption
- Cryptography Constants Detection
- Cipher Might Be Operating In ECB Mode

Other
- Command

Privacy
- Privacy Concern
- Clear Text Storage of sensitive information
- Expose sensitive information to Unauthorized
- Sensitive Information

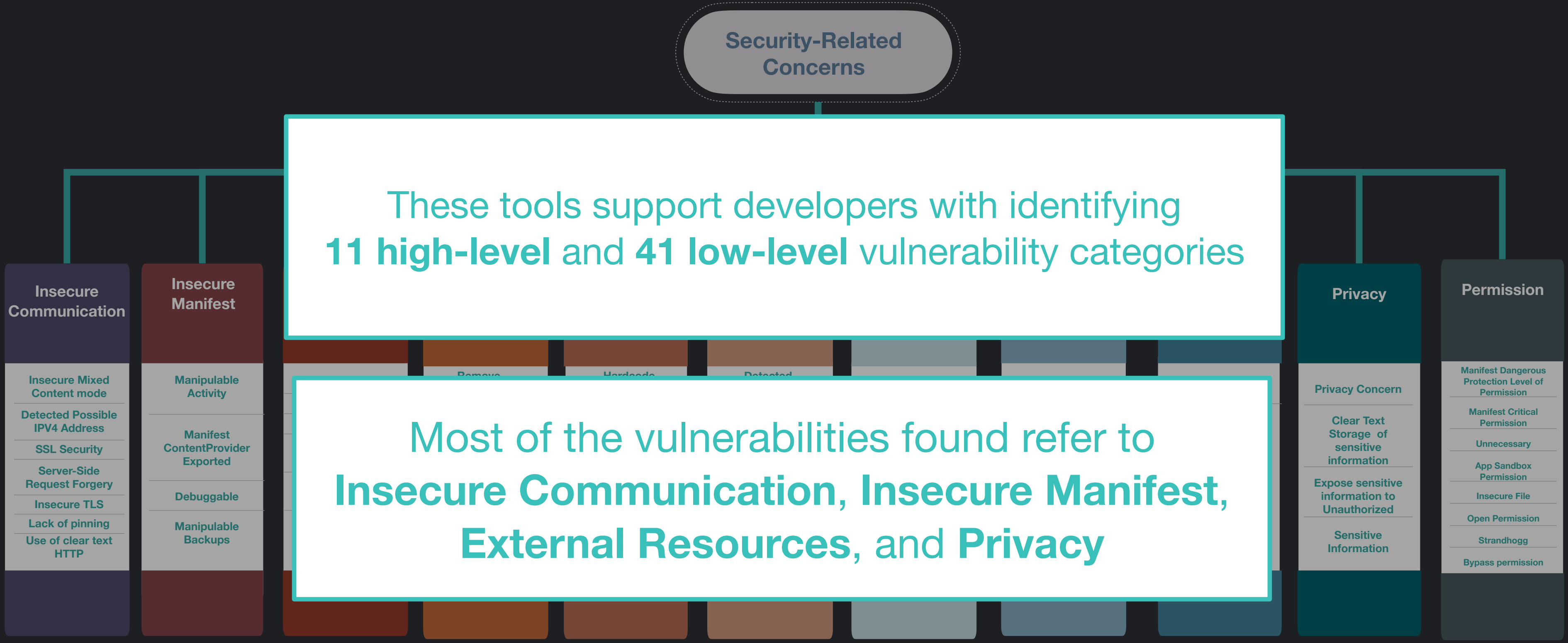
Permission
- Manifest Dangerous Protection Level of Permission
- Manifest Critical Permission
- Unnecessary
- App Sandbox Permission
- Insecure File
- Open Permission
- Strandhogg
- Bypass permission



Security-Related Concerns

These tools support developers with identifying **11 high-level** and **41 low-level** vulnerability categories

- Insecure Communication
  - Insecure Mixed Content mode
  - Detected Possible IPV4 Address
  - SSL Security
  - Server-Side Request Forgery
  - Insecure TLS
  - Lack of pinning
  - Use of clear text HTTP
- Insecure Manifest
  - Manipulable Activity
  - Manifest ContentProvider Exported
  - Debuggable
  - Manipulable Backups
- WebView
  - Detected Format
  - Detected Library
  - Detected possible FQDN
  - Detected path component
  - Detected URL
- Remove Android Device Lock by Rouge app
  - KeyStore
- Hardcode Certificates
- Detected Offuscator
  - Lack of Obfuscation
- Detected Logging
  - External storage Accessing
  - Database
- Base64 String Encryption
  - Cryptography Constants Detection
  - Cipher Might Be Operating In ECB Mode
- Command
- Privacy
  - Privacy Concern
  - Clear Text Storage of sensitive information
  - Expose sensitive information to Unauthorized
  - Sensitive Information
- Permission
  - Manifest Dangerous Protection Level of Permission
  - Manifest Critical Permission
  - Unnecessary
  - App Sandbox Permission
  - Insecure File
  - Open Permission
  - Strandhogg
  - Bypass permission



Security-Related Concerns

Insecure Communication
- Insecure Mixed Content mode
- Detected Possible IPV4 Address
- SSL Security
- Server-Side Request Forgery
- Insecure TLS
- Lack of pinning
- Use of clear text HTTP

Insecure Manifest
- Manipulable Activity
- Manifest ContentProvider Exported
- Debuggable
- Manipulable Backups

Remove

Hardcode

Detected

Privacy
- Privacy Concern
- Clear Text Storage of sensitive information
- Expose sensitive information to Unauthorized
- Sensitive Information

Permission
- Manifest Dangerous Protection Level of Permission
- Manifest Critical Permission
- Unnecessary
- App Sandbox Permission
- Insecure File
- Open Permission
- Strandhogg
- Bypass permission

These tools support developers with identifying **11 high-level** and **41 low-level** vulnerability categories

Most of the vulnerabilities found refer to **Insecure Communication**, **Insecure Manifest**, **External Resources**, and **Privacy**

Key findings of RQ1 - Vulnerabilities Identified by Tools

Category	Tools
Improper Platform Usage	Androbugs Trueeseeing
Insecure Data Storage	Androbugs Trueeseeing
Insecure Communication	Androbugs Insider Trueeseeing
Insufficient Authentication	Androbugs Trueeseeing
Insufficient Cryptography	Trueseeing
Insecure Authorization	\
Client Code Quality	\
Code tampering	Trueeseeing
Reverse Engineering	\
Extraneous Functionality	\

Key findings of RQ1 - Vulnerabilities Identified by Tools

OWASP Mobile Top-10 2016

Category	Tools
Improper Platform Usage	Androbugs Trueeseeing
Insecure Data Storage	Androbugs Trueeseeing
Insecure Communication	Androbugs Insider Trueeseeing
Insufficient Authentication	Androbugs Trueeseeing
Insufficient Cryptography	Trueseeing
Insecure Authorization	\
Client Code Quality	\
Code tampering	Trueeseeing
Reverse Engineering	\
Extraneous Functionality	\

Key findings of RQ1 - Vulnerabilities Identified by Tools

OWASP M
Top-10 2016

Category	Tools
Improper Platform Usage	Androbugs Trueeseeing
Insufficient Cryptography	Trueseeing
Insecure Authorization	\
Client Code Quality	\
Code tampering	Trueeseeing
Reverse Engineering	\
Extraneous Functionality	\

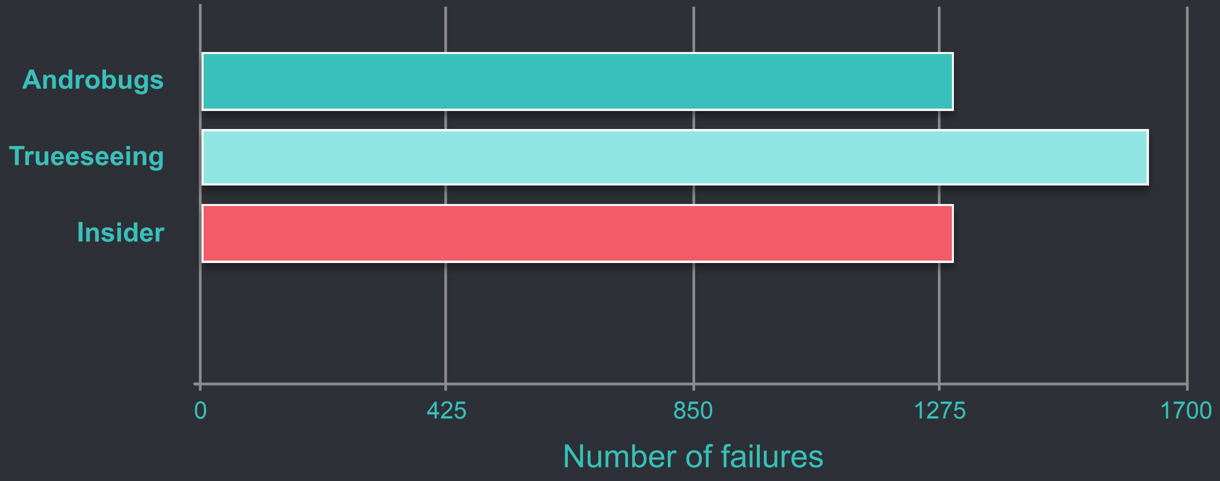
These tools only partially cover
the top-10 risks by OWASP

Research Questions

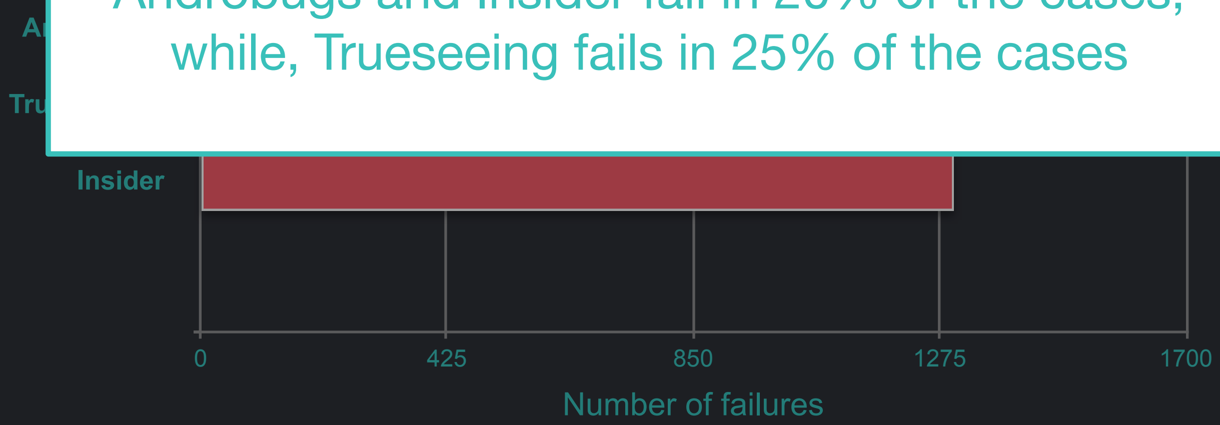
RQ1 - What are the **vulnerability types** identified by existing automated static analysis tools for mobile apps?

RQ2 - What are the capabilities of existing automated static analysis tools in terms of mobile **app analyzability**, **frequency of detection**, and **complementarity** among them?

RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?

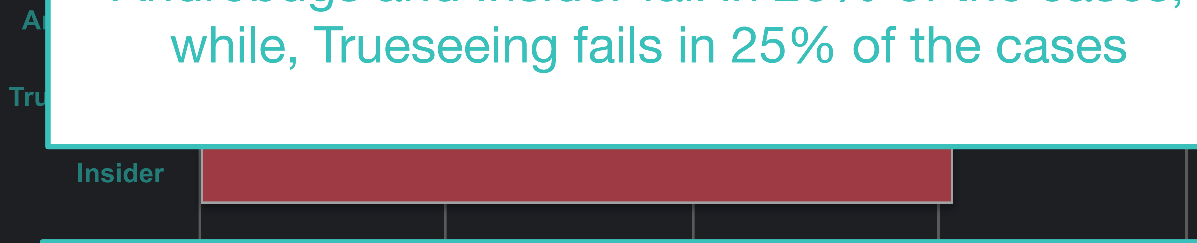


RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?



Androbugs and Insider fail in 20% of the cases, while, Trueseeing fails in 25% of the cases

RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?

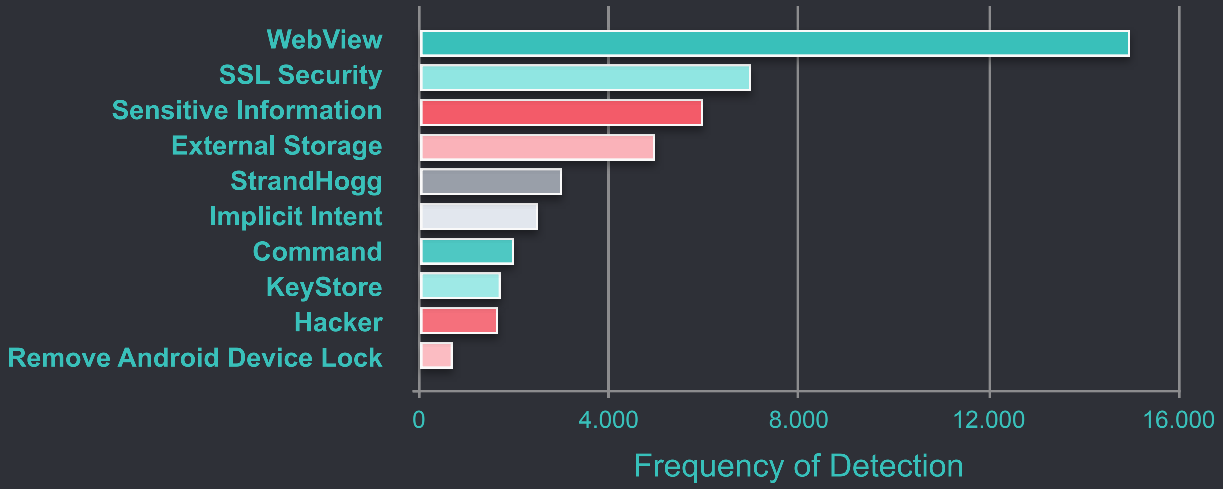


Androbugs and Insider fail in 20% of the cases, while, Trueseeing fails in 25% of the cases

These tools typically fail due to misconfiguration and wrong dependencies usage

RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?

Androbugs



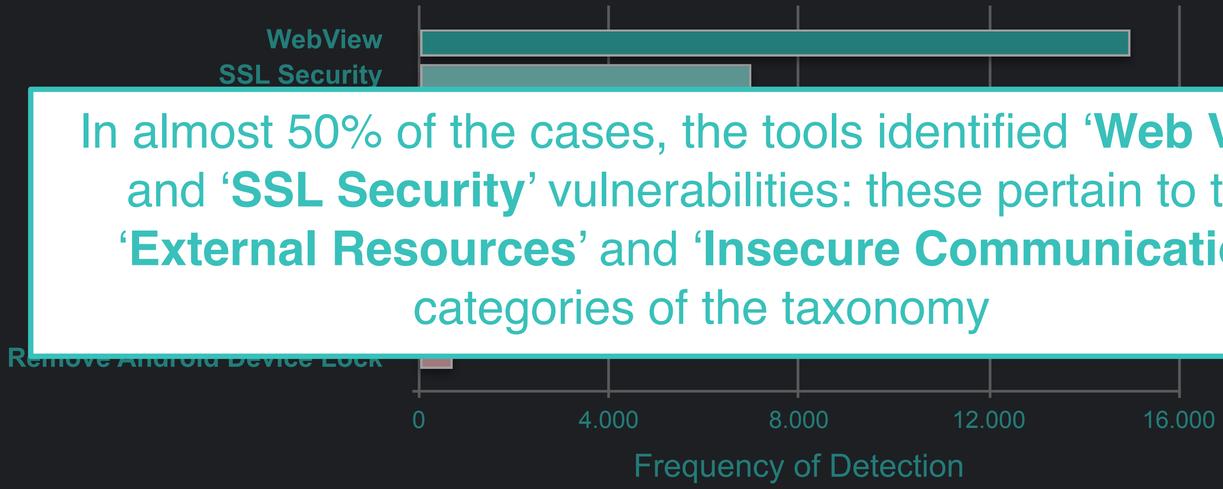
RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?

WebView

Developers require an external webpage and a malicious user could inject malicious code using JavaScript malicious components inside the webpage

RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?

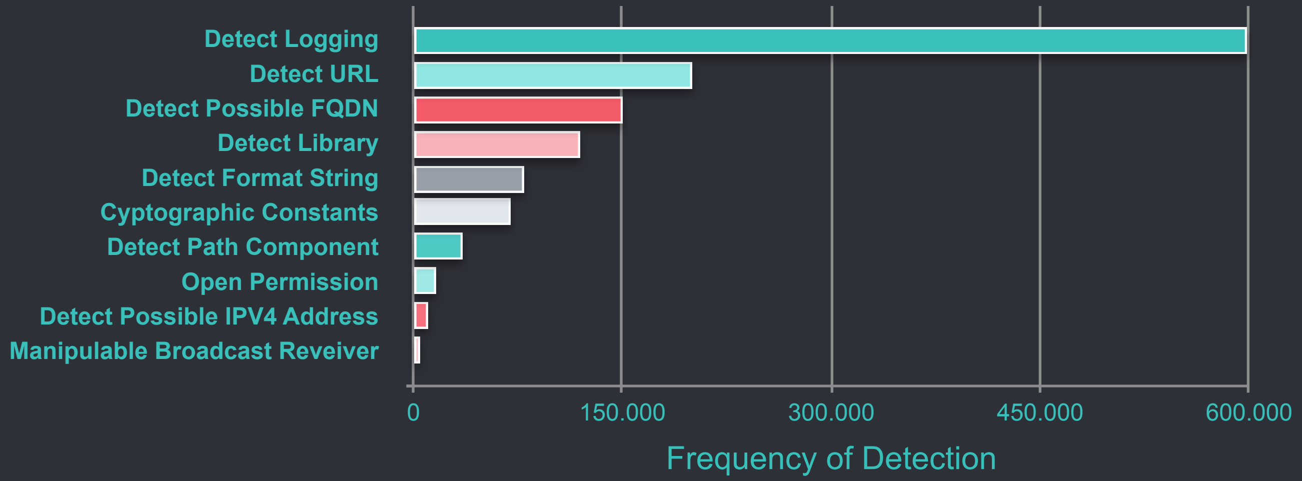
Androbugs



In almost 50% of the cases, the tools identified ‘**Web View**’ and ‘**SSL Security**’ vulnerabilities: these pertain to the ‘**External Resources**’ and ‘**Insecure Communication**’ categories of the taxonomy

RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?

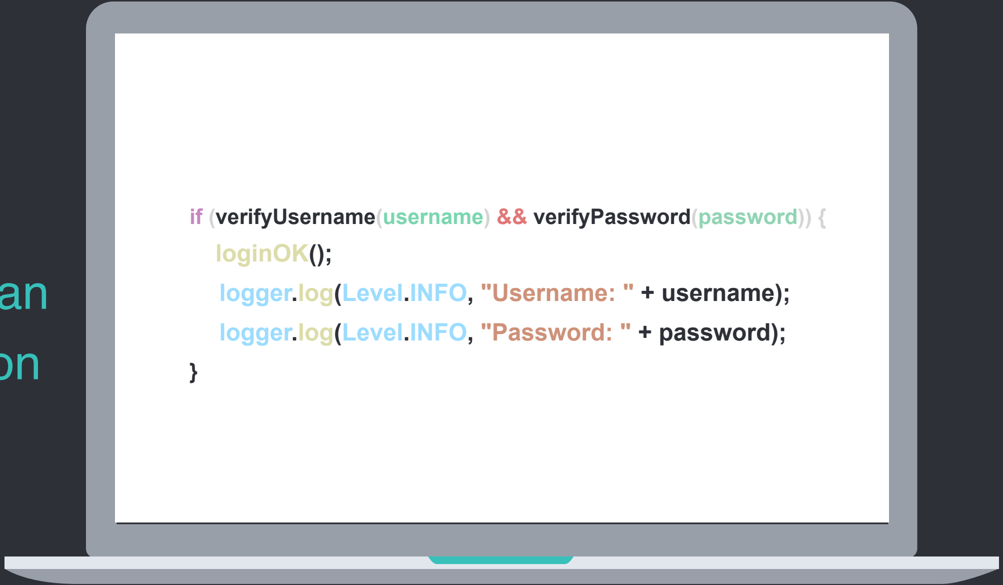
Trueeseeing



RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?

Detect Logging file

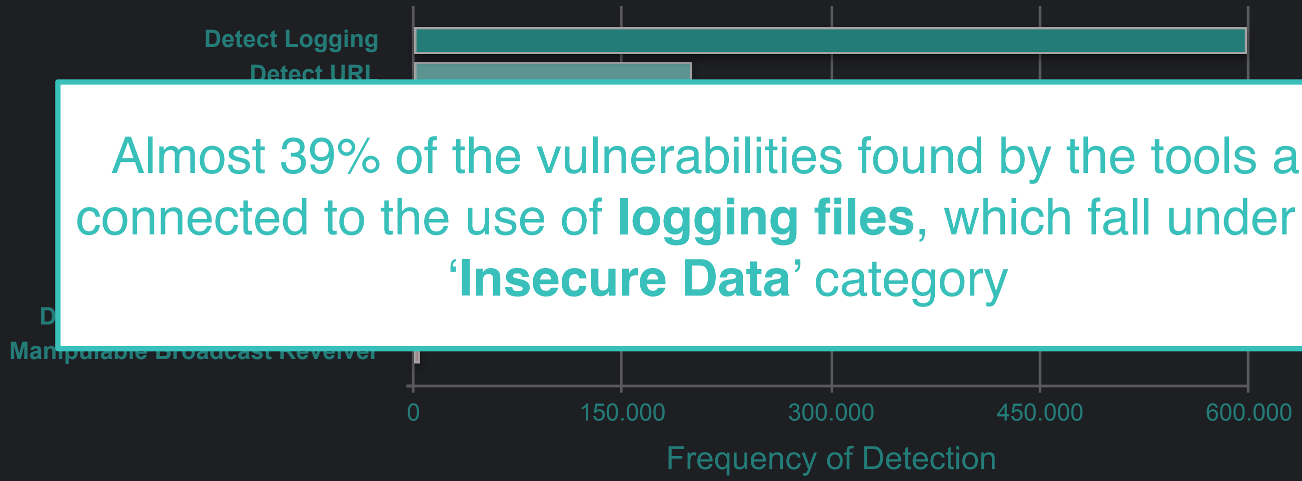
Developers could accidentally write sensitive information in a log file, and an attacker could identify these information to try an attack



```
if (verifyUsername(username) && verifyPassword(password)) {
   loginOK();
    logger.log(Level.INFO, "Username: " + username);
    logger.log(Level.INFO, "Password: " + password);
}
```

RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?

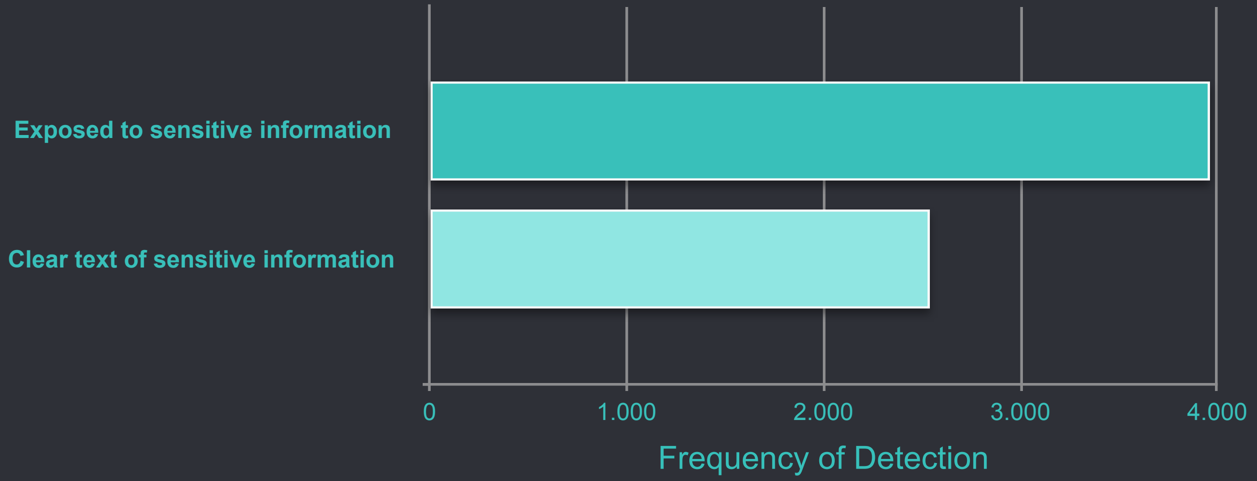
Trueeseeing



Almost 39% of the vulnerabilities found by the tools are connected to the use of **logging files**, which fall under the ‘**Insecure Data**’ category

RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?

Insider

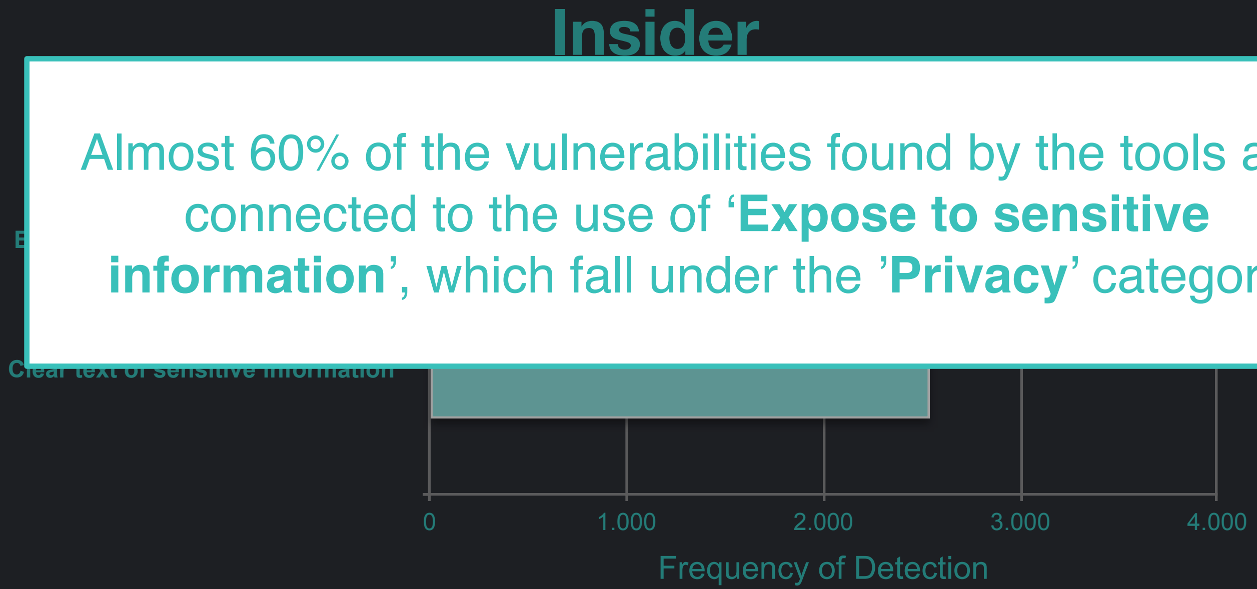


RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?

Exposed to sensitive information

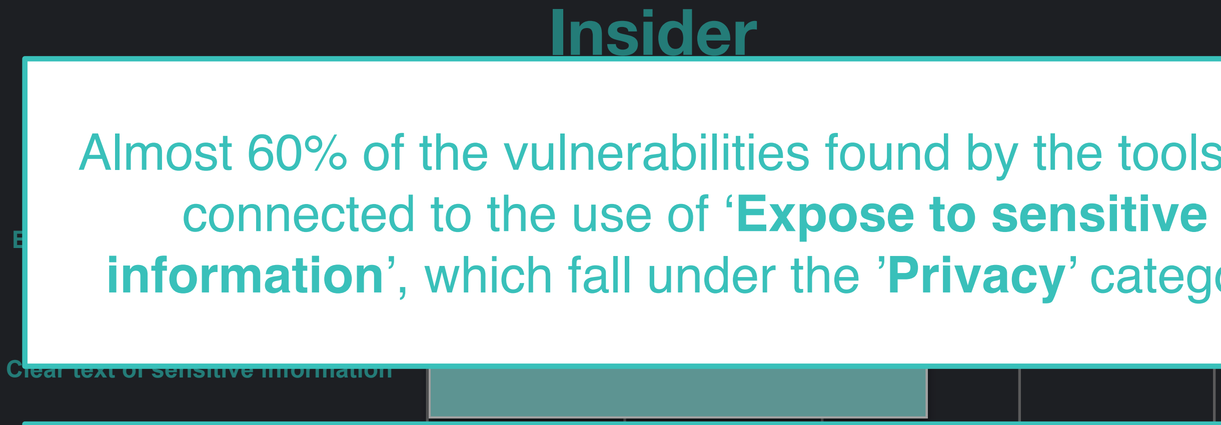
This vulnerability occurs when the developer does not use protection mechanisms appropriately when sharing or saving sensitive information

RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?



Almost 60% of the vulnerabilities found by the tools are connected to the use of ‘**Expose to sensitive information**’, which fall under the ’**Privacy**’ category

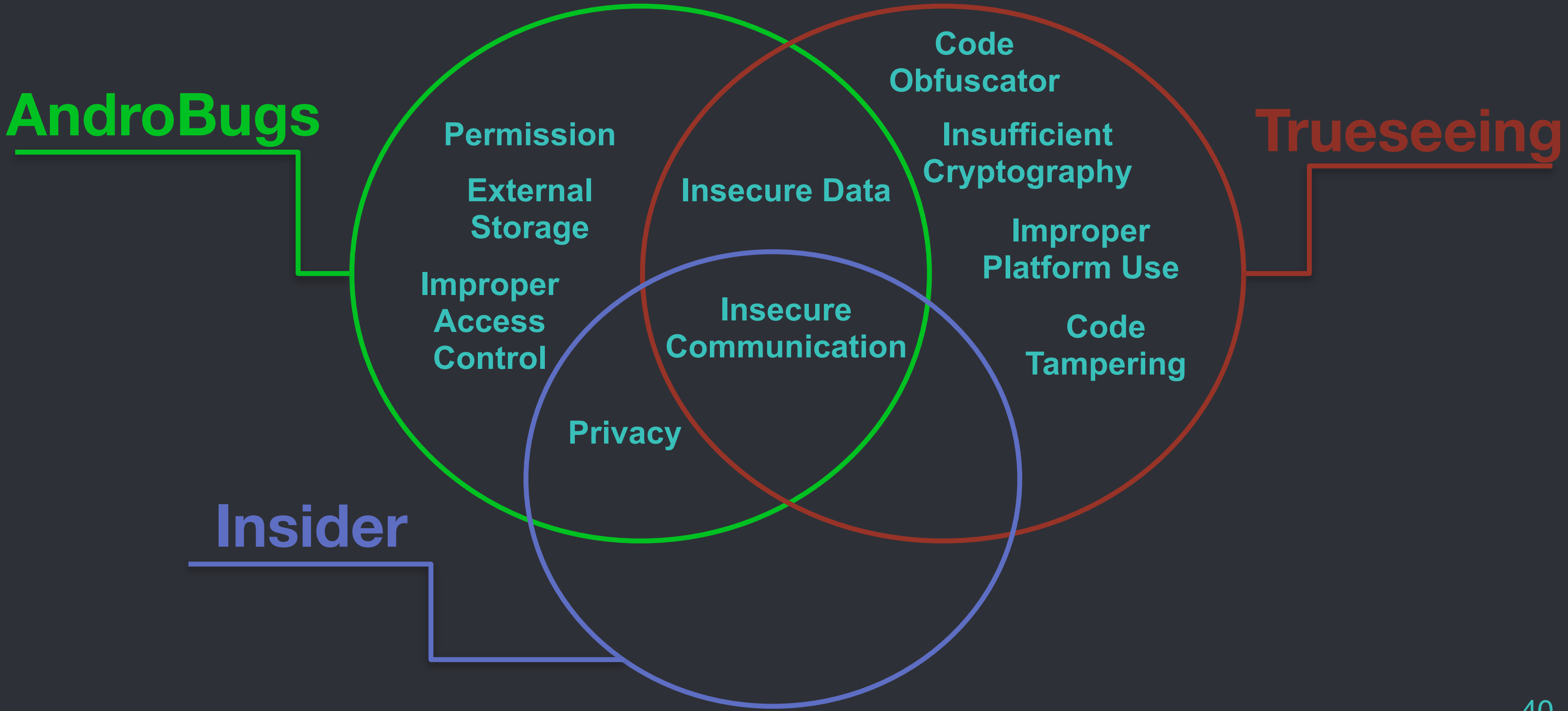
RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?



Almost 60% of the vulnerabilities found by the tools are connected to the use of ‘**Expose to sensitive information**’, which fall under the ’**Privacy**’ category

Although according to the official documentation, the tool can detect each vulnerability on the OWASP top 10. We observed a **partial mismatch** between the documentation and the real vulnerability detected

RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?



Key findings of RQ2 - Frequency

Key findings of RQ2 - Frequency

Different tools can detect different security-related concerns with different frequencies

Key findings of RQ2 - Frequency

Different tools can detect different security-related concerns with different frequencies

There are **vulnerabilities almost never detected** by these tools (e.g., Improper Access Control)

Key findings of RQ2 - Frequency

Different tools can detect different security-related concerns with different frequencies

There are **vulnerabilities almost never detected** by these tools (e.g., Improper Access Control)

A deeper analysis of the actual support provided by these tools could be necessary

Key findings of RQ2 - Complementarity

Key findings of RQ2 - Complementarity

AndroBugs and Trueseeing can cover different security-related problems, suggesting a sort of complementarity between them

Key findings of RQ2 - Complementarity

AndroBugs and Trueseeing can cover different security-related problems, suggesting a sort of complementarity between them

Insider can detect **only a subset** of vulnerabilities also detected by Androbug and Trueseeing

Summing up

The results obtained indicate that:

Summing up

The results obtained indicate that:

- The selected tools can detect 11 high-level vulnerabilities categories and 41 low-level ones

Summing up

The results obtained indicate that:

- The selected tools can detect 11 high-level vulnerabilities categories and 41 low-level ones
- The selected tools only partially cover the top-10 risks by OWASP

Summing up

The results obtained indicate that:

- The selected tools can detect 11 high-level vulnerabilities categories and 41 low-level ones
- The selected tools only partially cover the top-10 risks by OWASP
- Practitioners should combine multiple tools to identify as many vulnerabilities as possible

Summing up

The results obtained indicate that:

- The selected tools can detect 11 high-level vulnerabilities categories and 41 low-level ones

- The selected tools only partially cover the top-10 risks by OWASP

- Practitioners should combine multiple tools to identify as many vulnerabilities as possible

Future Work

Summing up

The results obtained indicate that:

- The selected tools can detect 11 high-level vulnerabilities categories and 41 low-level ones
- The selected tools only partially cover the top-10 risks by OWASP
- Practitioners should combine multiple tools to identify as many vulnerabilities as possible

Future Work

- Manual evaluation of the accuracy of selected static analysis tools

Summing up

The results obtained indicate that:

- The selected tools can detect 11 high-level vulnerabilities categories and 41 low-level ones
- The selected tools only partially cover the top-10 risks by OWASP
- Practitioners should combine multiple tools to identify as many vulnerabilities as possible

Future Work

- Manual evaluation of the accuracy of selected static analysis tools
- Expand the study by including other tools (e.g., machine learning tools)

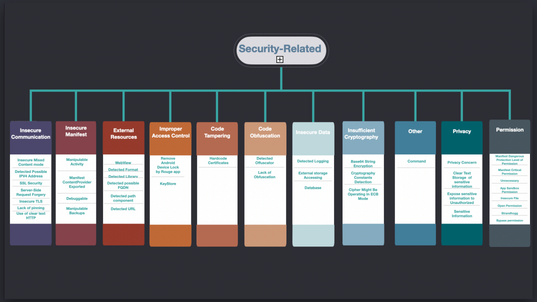
Summing up

The results obtained indicate that:

- The selected tools can detect 11 high-level vulnerabilities categories and 41 low-level ones
- The selected tools only partially cover the top-10 risks by OWASP
- Practitioners should combine multiple tools to identify as many vulnerabilities as possible

Future Work

- Manual evaluation of the accuracy of selected static analysis tools
- Expand the study by including other tools (e.g., machine learning tools)
- Expand the dataset to include paid applications



Security-Related

Insecure Communication
Insecure Mixed Content mode
Detected Possible IPV4 Address
SSL Security
Server-Side Request Forgery
Insecure TLS
Lack of pinning
Use of clear text HTTP

Insecure Manifest
Manipulable Activity
Manifest ContentProvider Exported
Debuggable
Manipulable Backups

External Resources
WebView
Detected Format
Detected Library
Detected possible FQDN
Detected path component
Detected URL

Improper Access Control
Remove Android Device Lock by Rouge app
KeyStore

Code Tampering
Hardcode Certificates

Code Obfuscation
Detected Offuscator
Lack of Obfuscation

Insecure Data
Detected Logging
External storage Accessing
Database

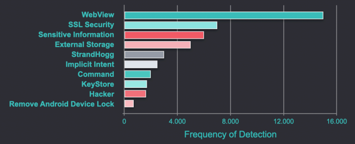
Insufficient Cryptography
Base64 String Encryption
Cryptography Constants Detection
Cipher Might Be Operating In ECB Mode

Other
Command

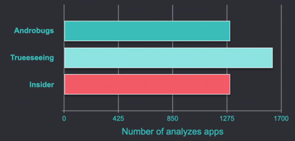
Privacy
Privacy Concern
Clear Text Storage of sensitive information
Expose sensitive information to Unauthorized
Sensitive Information

Permission
Manifest Dangerous Protection Level of Permission
Manifest Critical Permission
Unnecessary
App Sandbox Permission
Insecure File
Open Permission
Strandhogg
Bypass permission

RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?



RQ2: What are the capabilities of existing automated static analysis tools in terms of mobile app analyzability, frequency of detection, and complementarity among them?



Summing up

The results obtained indicate that:

- 👍 The selected tools can detect 11 high - level vulnerabilities categories and 41 low - level ones
- 👍 The selected tools only partially cover the top 10 vulnerabilities listed by OWASP
- 👍 Practitioners should combine multiple tools to identify as many vulnerabilities as possible

Our replication package is available here:

Scan me!

sesa lab

SOFTWARE ENGINEERING SALERNO

Scan me!

giagiordano@unisa.it

https://broke31.github.io/giammaria-giordano/

GiammariaGiord1